Annotated Bibliography

Acknowledgement: The starting point for this bibliography was the extensive annotated bibliography found in Chapter 12 of Gary McGraw's book *Software Security: Building Security In*[1]. This made the bibliography much easier to develop than if we had started from scratch.

Alberts, Christopher J. & Dorofee, Audrey J. *Managing Information Security Risks: The OCTAVE*^{SM} *Approach*. Boston, MA: Addison-Wesley, 2002 (ISBN 0321118863).

This is a descriptive and process-oriented book on a new security risk evaluation method, OCTAVE. OCTAVE stands for Operationally Critical Threat, Asset, and Vulnerability Evaluation. An information security risk evaluation helps organizations evaluate organizational practice as well as the installed technology base and to make decisions based on potential impact. (from the WorldCat Database)

Allen, Julia H. *The CERT Guide to System and Network Security Practices*. Boston, MA: Addison-Wesley, 2001 (ISBN 020173723X).

Showing how to improve system and network security, this guide explores the practices and policies of deploying firewalls, securing network servers, securing desktop workstations, intrusion detection, response, and recovery. (from the WorldCat Database)

Amoroso, Ed. *Fundamentals of Computer Security Technology*. Englewood Cliffs, NJ: Prentice Hall, 1994 (ISBN 0131089293).

Tutorial in style, this volume provides a comprehensive survey of the state of the art of the entire field of computer security. It first covers the threats to computer systems and then discusses all the models, techniques, and mechanisms designed to thwart those threats, as well as known methods of exploiting vulnerabilities. (from the WorldCat Database)

Anderson, Ross. *Security Engineering: A Guide to Building Dependable Distributed Systems*[2]. New York, NY: John Wiley and Sons, 2001 (ISBN 0471389226).

Presents an accessible discussion of security engineering basics, from protocols to distributed systems, and explains protection technologies such as biometrics, tamper resistance, security seals, and copyright marketing. Describes what sort of attacks are done on a range of systems, including banking, medical records, and e-commerce, and tells how to stop attacks. There is also discussion of how computer security interacts with the law and with corporate culture. Anderson directs research in computer security at Cambridge University, England. (Annotation © Book News, Inc., Portland, OR; booknews.com)

Berg, Clifford J. *High-Assurance Design : Architecting Secure and Reliable Enterprise Applications*. Boston, MA: Addison-Wesley, 2005 (ISBN 0321375777).

Cliff Berg shows how to design high-assurance applications that build in reliability, security, manageability, and maintainability up front. He draws on real-world scenarios and actual applications, focusing heavily on the activities and relationships associated with building superior software. (powells.com)

Bishop, Matt. *Computer Security: Art and Science*. Boston, MA: Addison-Wesley, 2003 (ISBN 0201440997).

This book fully introduces the theory and practice of computer security. It is both a comprehensive text,

1. daisy:523#ss (Annotated Bibliography)
2. http://www.cl.cam.ac.uk/~rja14/book.html

explaining the most fundamental and pervasive aspects of the field, and a detailed reference…The author incorporates concepts from computer systems, networks, human factors, and cryptography. In doing so, he effectively demonstrates that computer security is an art as well as a science. (from the WorldCat Database)

Brooks, Frederick Jr. *The Mythical Man-Month: Essays on Software Engineering, 2nd ed*. Reading, MA: Addison-Wesley, 1995 (ISBN 0201835959).

No book on software project management has been so influential and so timeless as The Mythical Man-Month. Now, 20 years after the publication of his book, Brooks revisits his original ideas and develops new thoughts and advice both for readers familiar with his work and for readers discovering it for the first time. (from the WorldCat Database)

Cheswick, Bill; Bellovin, Steve; & Rubin, Avi. *Firewalls and Internet Security, 2nd ed*. Boston, MA: Addison-Wesley, 2003 (ISBN 020163466X).

Offers a step-by-step approach to planning, designing, and implementing a complete security strategy that can thwart sophisticated hackers while allowing a company easy access to their own Internet services. (from the WorldCat Database)

Committee on Information Systems Trustworthiness, National Research Council. *Trust in Cyberspace*[3]. Edited by Fred Schneider. Washington, DC: National Academy Press, 1999 (ISBN 0309065585).

This book provides an assessment of the current state of the art for building trustworthy networked information systems. It proposes directions for research in computer and network security, software technology, and system architecture. In addition, it assesses current technical and market trends in order to better inform public policy as to where progress is likely and where incentives could help. (from the WorldCat Database)

Coplien, James O. & Harrison, Neil B. *Organizational Patterns of Agile Software Development*. Upper Saddle River, NJ: Prentice Hall, 2004 (ISBN 0131467409).

This book covers the human and organizational dimension of the software improvement process and software project management—whether based on the CMM or ISO 9000 or the Rational Unified Process. Drawn from a decade of research, it emphasizes common-sense practices. It describes four "pattern languages": how to manage a project, how to grow it over time, what can make up an organization's "style," and how the people fulfill their roles and interact with each other. These are not prescriptions or algorithms; they' re elements of how successful organizations have worked. Historical supporting material from other disciplines is provided. (from Slashdot)

Denning, Dorothy. *Information Warfare and Security*. Reading, MA: Addison-Wesley, 1998 (ISBN 0201433036).

Security expert Dorothy Denning focuses on the criminals and information terrorists whose depredations include information-based threats to nations, corporations, and individuals. From government use of information warfare for law enforcement investigations and military and intelligence operations to conflicts arising in the areas of free speech and encryption, this book places cybercrime within a broader context, integrating the various kinds of information crime—and the countermeasures against it—into a methodology-based framework. The approach addresses offensive information warfare (including acquisition of information) deceptive exploitation of information, and denial of access to information. Additionally, Denning presents case examples, including the Persian Gulf War, stressing actual incidents to illustrate instances of information warfare. (from the WorldCat Database)

Farmer, Dan & Venema, Wietse. *Forensic Discovery*. Boston, MA: Addison-Wesley, 2005 (ISBN

3. http://www.nap.edu/readingroom/books/trust/

020163497X).

Computer forensics—the art and science of gathering and analyzing digital evidence, reconstructing data and attacks, and tracking perpetrators—is becoming ever more important as IT and law enforcement professionals face an epidemic in computer crime. In Forensic Discovery, two internationally recognized experts present a thorough and realistic guide to the subject. The authors draw on their extensive firsthand experience to cover everything from file systems to memory and kernel hacks to malware. Readers will find extensive examples from Solaris, FreeBSD, Linux, and Microsoft Windows, as well as practical guidance for writing one's own forensic tools.

The book's companion Web site contains complete source and binary code for open source software discussed in the book, plus additional computer forensics case studies and resource links. (from the WorldCat Database)

Ford, Warwick. *Computer Communications Security: Principles, Standard Protocols, and Techniques*. Englewood Cliffs, NJ: Prentice Hall, 1994 (ISBN 0137994532).

For anyone required to design, develop, implement, market, or procure products based on specific network security standards, this book identifies and explains all the modern standardized methods of achieving network security in both TCP/IP and OSI environments--with a focus on inter-system, as opposed to intra-system, security functions. (from the WorldCat Database)

Gamma, Erich; Helm, Richard; Johnson, Ralph; & Vlissides, John. *Design Patterns*. Reading, MA: Addison-Wesley, 1995 (ISBN 0201633612).

[The] authors present the first book containing a catalog of object-oriented design patterns. Readers can learn how to use design patterns in the object-oriented development process, solve specific design problems using patterns, and gain a common vocabulary for object-oriented design. (from the WorldCat Database)

Garfinkel, Simson & Spafford, Gene. *Practical UNIX and Internet Security, 2nd ed*. Sebastopol, CA: O'Reilly, 1996 (ISBN 1565921488).

Covers Internet security and networking issues, including World Wide Web security, wrapper and proxy programs, integrity management tools, secure programming, and how to secure TCP/IP services (e.g., FTP, SMTP, DNS). Chapters on host security contain details on passwords, the UNIX file system, cryptography, backups, logging, physical security, telephone security, UUCP, firewalls, and dealing with breakins. Includes summary appendixes on freely available security tools, references, and security-related organizations. (from the WorldCat Database)

Gasser, Morrie. *Building a Secure Computer System*[4]. New York, NY: Van Nostrand Reinhold, 1988 (ISBN 0442230222).

A very old but interesting read that anticipates the philosophy of building security in some twenty years earlier. (from Gary McGraw's book *Software Security: Building Security In*)

Gilb, Tom & Graham, Dorothy. *Software Inspection*. Reading, MA: Addison-Wesley, 1993 (ISBN 0201631814).

Gilb and Graham show software professionals how to achieve high-quality software through inspection. They show how to do a formal review of documents to find errors, giving effective statistical process improvement. The book includes many examples and case studies based on actual experience at IBM, AT&T, McDonnell Douglas, and other companies. (from the WorldCat Database)

Gollmann, Dieter. *Computer Security, 2nd ed*. New York, NY: John Wiley & Sons, 2006 (ISBN

4. http://nucia.ist.unomaha.edu/library/gasserbook.pdf

0470862939).

This is…a publication on the technical aspects of computer security. Derived from a one-year post graduate course on information security, it discusses fundamental concepts, the application of security to current systems (UNIX and NT), distributed system security and the theoretical basis of database and multi-level security. Familiarity with operating systems, applications and security concepts at an advanced level is assumed. The author is associated with Microsoft Research Ltd, Cambridge, UK. (from barnesandnoble.com)

Graff, Mark & van Wyk, Kenneth. *Secure Coding: Principles and Practices*. Sebastopol, CA: O'Reilly and Associates, 2003 (ISBN 0596002424).

Sheds light on the economic, psychological, and practical reasons why security vulnerabilities are so ubiquitous today. (from the WorldCat Database)

Hoglund, Greg & Butler, James. *Rootkits: Subverting the Windows Kernel*. Boston, MA: Addison-Wesley, 2005 (ISBN 0321294319).

Assuming a familiarity with C and Windows device driver architecture, this guide describes the generic approaches used by rootkits to invade computer systems and remain there undetected. It covers userland and kernel hooks, runtime patching, keyboard sniffers, direct kernel object manipulation, and covert channels. (Annotation © 2005 Book News, Inc., Portland, OR)

Hoglund, Greg & McGraw, Gary. *Exploiting Software: How to Break Code*[5]. Boston, MA: Addison-Wesley, 2004 (ISBN: 0201786958).

Intended for software security professionals, this guide explains the techniques used by malicious hackers against software, describes specific attack patterns, and shows how to uncover new software vulnerabilities. The authors discuss the difference between implementation bugs and architectural flaws, reverse engineering tools, the weaknesses in server and client software, malicious input attacks, buffer overflows, and the construction of a simple Windows XP kernel rootkit that can hide processes and directories. (Annotation © 2004 Book News, Inc., Portland, OR)

Howard, Michael[6] & LeBlanc, David. *Writing Secure Code, 2nd ed*. Redmond, WA: Microsoft Press, 2003 (ISBN 0735617228).

Howard and LeBlanc (both are security experts with Microsoft) discuss the need for security and outline its general principles before outlining secure coding techniques. Testing, installation, documentation, and error messages are also covered. Appendices discuss dangerous APIs, dismiss pathetic excuses, and provide security checklists. The book explains how systems can be attacked, uses anecdotes to illustrate common mistakes, and offers advice on making systems secure. (Annotation © Book News, Inc., Portland, OR; booknews.com)

Howard, Michael; LeBlanc, David; & Viega, John. *19 Deadly Sins of Software Security*. New York, NY: McGraw-Hill Osborne Media, 2005 (ISBN 0072260858).

This book outlines the 19 sins of software security and shows how to fix each one. Detailed code examples throughout the book show the code defects as well as the fixes and defenses. (Adapted from SearchWindowsSecurity.com)

Howard, Michael & Lipner, Steve. *The Security Development Lifecycle*. Redmond, WA: Microsoft Press, 2003 (ISBN 0735622140).

Howard and Lipner describe Microsoft's process for attempting to reduce the number of security defects

5. http://www.exploitingsoftware.com

6. http://blogs.msdn.com/michael_howard/

in code at every stage of the development process. Reference material includes information about integrating SDL with agile methods, banned function calls, minimum cryptographic standards, required tools and compiler options, and threat tree patterns.

Kahn, David. *The Code-Breakers* (revised edition). New York, NY: Scribner, 1996 (ISBN 0684831309).

A readable, lucid introduction to encryption with an emphasis on WWII applications, but with a range from Caesar to PGP. (from Slashdot)

Kaner, Cem & Pels, David. *Bad Software: What to Do When Software Fails*. New York, NY: John Wiley & Sons, 1998 (ISBN 0471318264).

The book unflinchingly explains why software companies have the upper hand and use it, and how to avoid getting stuck with a lemon. The authors provide techniques to use in dealing with vendor or consumer protection agencies. Key coverage includes troubleshooting software; what to ask for when defective products cost you time and money; and your rights and how to enforce them. (from the WorldCat Database)

Kernighan, Brian & Ritchie, Dennis. *The C Programming Language, 2nd ed*. New York, NY: Prentice Hall, 1988 (ISBN 0131103709).

Introduces the features of the C programming language, discusses data types, variables, operators, control flow, functions, pointers, arrays, and structures, and looks at the UNIX system interface. (from the WorldCat Database)

Knuth, Donald. *The Art of Computer Programming: Seminumerical Algorithms, 3rd ed*. Reading, MA: Addison-Wesley, 1997 (ISBN 0201896834).

Covers information structures—the representation of information within a computer, the structural interrelations between data elements and how to work with them efficiently, and applications to simulation, numerical methods, and software design. (from the WorldCat Database)

Koziol, Jack; Litchfield, David; Aitel, Dave; Anley, Chris; Eren, Sinan “noir”; Mehta, Neel; & Hassell, Riley. *The Shellcoder’s Handbook: Discovering and Exploiting Security Holes*. New York, NY: John Wiley & Sons, 2004 (ISBN 0764544683).

Covers basic vulnerability discovery and development on popular operating systems and applications. Also includes detailed instructions on putting together exploits for discovered vulnerabilities, as well as some advanced techniques that are not public knowledge. (from powells.com)

LaMacchia, Brian; Lang, Sebastian; Lyons, Matther; Martin, Rui; & Price, Kevin. *.NET Framework Security*. Boston, MA: Addison-Wesley, 2002 (ISBN 067232184X).

.NET Framework Security is the authoritative, comprehensive, technical guide to the security features of the .NET platform written by the Microsoft developers who are writing, testing, and executing those features…Includes anecdotes, tips, and "what if" scenarios based on Microsoft developers' actual experiences, both past and present.

.NET Framework Security contains security features of the .NET Framework and Common Language Runtime. The focus is on the new technologies introduced with .NET, with treatment of existing technologies as leveraged by .NET (but not those existing technologies themselves). (from the WorldCat Database)

Leveson, Nancy. Safeware: *System Safety and Computers*. Reading, MA: Addison-Wesley, 1995 (ISBN 0201119722).

Software engineers and system developers need to understand the issues and develop the skills required to prevent destructive accidents before they occur. This book examines what is currently known about

building safe electromechanical systems and looks at past accidents to see what lessons can be applied to new computer-controlled systems. (from the WorldCat Database)

Maguire, Steve. *Writing Solid Code*. Redmond, WA: Microsoft Press, 1993 (ISBN 1556155514).

A Microsoft developer examines the problem of programming "bugs," showing how and where developers make mistakes along the development process and providing ways users can detect errors early. (from the WorldCat Database)

McClure, Stuart; Scambray, Joel; & Kurtz, George. *Hacking Exposed: Network Security Secrets and Solutions, 5th ed*. New York, NY: Osborne/McGraw-Hill, 2005 (ISBN 0072260815).

Using real-world case studies, renowned security experts Stuart McClure, Joel Scambray, and George Kurtz show IT professionals how to protect computers and networks against the most recent security vulnerabilities. You'll find detailed examples of the latest devious break-ins and will learn how to think like a hacker in order to thwart attacks. Coverage includes code hacking methods and countermeasures, new exploits for Windows 2003 Server, UNIX/Linux, Cisco, Apache, and Web and wireless applications, latest DDoS techniques, new classes of vulnerabilities. (from barnesandnoble.com)

McGraw, Gary. *Software Security: Building Security In*. Upper Saddle River, NJ : Addison-Wesley, 2006 (ISBN 0321356705).

Describes how to put software security into practice, covering such topics as risk management frameworks, architectural risk analysis, security testing, and penetration testing. (from the WorldCat Database)

McGraw, Gary & Felten, Edward. *Java Security: Hostile Applets, Holes, and Antidotes*. New York, NY: John Wiley & Sons, 1996 (ISBN 047117842X).

Java Security outlines secure programming practices for Java. They reveal the weaknesses and pitfalls of current safe Java policy and show how to incorporate both organizational and technical fixes into an effective safety management program. (from barnesandnoble.com)

McGraw, Gary & Felten, Edward. *Securing Java: Getting Down to Business with Mobile Code*[7]. New York, NY: John Wiley & Sons, 1999 (ISBN 047131952X).

McGraw and Felten describe the taxonomy of nearly every recorded Java security lapse, whether inherent in Sun’s design or resultant from vendor miscues in virtual machine implementation. While many of the holes in the model have already been patched, the emphasis is on what types of things to look for, from what directions one might anticipate finding a security hole…The appendices cover, among other things, URLs to sites relevant to Java security and a long list of frequently asked questions about Java security. (from review by Jack Woehr, Dr. Dobb’s Electronic Review of Computer Books)

Menezes, Alfred; van Oorschot, Paul; & Vanstone, Scott. *Handbook of Applied Cryptography*[8]. CRC Press, 1996 (ISBN 0849385237).

A hefty handbook for both novices and experts, introducing practical aspects of conventional and public-key cryptography and offering information on the latest techniques and algorithms in the field. Mathematical treatments accompany practical discussions of areas including pseudorandom bits and sequences, stream and block ciphers, hash functions, and digital signatures. Also covers establishment protocols, implementation, and patents and standards. Includes annotated chapter references, and cross-referencing between chapters, plus a bibliography of papers from selected cryptographic forums. (Annotation © Book News, Inc., Portland, Or.)

7. http://www.securingjava.com/

8. http://www.cacr.math.uwaterloo.ca/hac/

Mouratidis, H. & Giorgini, Paolo, eds. *Integrating Security and Software Engineering: Advances and Future Vison*. Hershey, PA: Idea Group Publishing, 2007 (ISBN 1599041472).

This book investigates the integration of security concerns into software engineering practices, drawing expertise from the security and the software engineering community, and discusses future visions and directions for the field of secure software engineering. (from the publisher)

Neumann, Peter. *Computer-Related Risks*. Reading, MA: Addison-Wesley, 1995 (ISBN 020155805X).

Summarizes many real events involving computer technologies and the people who depend on those technologies, with widely ranging causes and effects. It considers problems attributable to hardware, software, people, and natural causes. Examples include disasters (such as the Black Hawk helicopter and Iranian Airbus shootdowns, the Exxon Valdez, and various transportation accidents); malicious hacker attacks; outages of telephone systems and computer networks; financial losses and many other strange happenstances (squirrels downing power grids, and April Fool's Day pranks). Computer-Related Risks addresses problems involving reliability, safety, security, privacy, and human well-being. It includes analyses of why these cases happened and discussions of what might be done to avoid recurrences of similar events. (from the WorldCat Database)

Rubin, Avi. *The Whitehat Security Arsenal: Tackling the Threats*. Reading, MA: Addison-Wesley, 1999 (ISBN 0201711141).

White Hats are the people doing good things with security, and this is their arsenal. The book covers everyday security issues and explains how to find the real threats and discover their solutions. (from the WorldCat Database)

Schneier, Bruce. *Applied Cryptography*. New York, NY: John Wiley & Sons, 1996 (ISBN 0471117099).

The guide to using cryptography to maintain data security, first published in 1994. It describes cryptography algorithms, gives advice on how to implement them into cryptographic software, and shows how they can be used to solve security problems. This edition covers protocols and algorithms to implement a variety of encryptions. (adapted from Annotation © Book News, Inc., Portland, OR; booknews.com)

Schneier, Bruce. *Secrets and Lies*. New York, NY: John Wiley & Sons, 2000 (ISBN 0471253111).

The author of Applied Cryptography lays out the realistic choices for those seeking security in a digital age, exploring various options and explaining the ins and outs of cryptography. (from the WorldCat Database)

Seacord, Robert C. *Secure Coding in C and C++*. Upper Saddle River, NJ: Addison-Wesley, 2006 (ISBN 0321335724).

Secure Coding in C and C++ provides…a detailed explanation of common programming errors in C and C++ and describes how these errors can lead to code that is vulnerable to exploitation. The book concentrates on security issues intrinsic to the C and C++ programming languages and associated libraries. It…is organized around functional capabilities commonly implemented by software engineers that have potential security consequences, such as formatted output and arithmetic operations. Each chapter describes insecure programming practices and common errors that can lead to vulnerabilities, how these programming flaws can be exploited, the potential consequences of exploitation, and secure alternatives. (from the publisher, Addison-Wesley)

Viega, John & McGraw, Gary. *Building Secure Software: How to Avoid Security Problems the Right Way*[9]. Boston, MA: Addison-Wesley, 2001 (ISBN 020172152X).

9. http://www.buildingsecuresoftware.com/

Though they include low-level detail that is most applicable to programmers, Viega and McGraw address anyone involved in software development from managers to coders. They explain how to build security into software at its very beginning. They refer readers to many other appropriate works for information on how to implement security measures on software and systems that already exist. (Annotation © Book News, Inc., Portland, OR; booknews.com)

Voas, Jeff & McGraw, Gary. *Software Fault Injection: Inoculating Programs against Errors*. New York, NY: John Wiley & Sons, 1998 (ISBN 0471183814).

Builds on the concepts presented in Friedman and Voas's book by looking at several real applications of fault injection. Introduces the terminology and definitions necessary for the application of this relatively new technology, offers several techniques for its implementation, and deals with the issues of software mutation and safety, information security, and maintenance and reuse. Concludes with a chapter on inoculating real- world software. The CD-ROM contains the SafetyNet fault injection tool with an accompanying HTML tutorial and Mothra, a software mutation tool. (Annotation © Book News, Inc., Portland, Or.)

Whittaker, James. *How to Break Software: A Practical Guide to Testing*. Boston, MA: Addison-Wesley, 2002 (ISBN 0201796198).

How to Break Software provides a practical tutorial on how to actually do testing by presenting numerous "attacks" you can perform to test your software for bugs. [It] is a departure from conventional testing in which testers prepare a written test plan and then use it as a script when testing the software. The testing techniques in this book are as flexible as conventional testing is rigid. And flexibility is needed in software projects in which requirements can change, bugs can become features and schedule pressures often force plans to be reassessed…Instead of a plan, intelligence, insight, experience and a "nose for where the bugs are hiding" should guide testers. This book helps testers develop this insight. This book does teach planning, but in an "on- the-fly while you are testing" way. It also encourages automation with many repetitive and complex tasks that require good tools (one such tool is shipped with this book on the companion CD). (from the publisher)

Whittaker, James & Thompson, Herbert. *How to Break Software Security*. Boston, MA: Addison-Wesley, 2003 (ISBN 0321194330).

Intended for software testers, this guide presents testing techniques that expose security holes caused by software dependencies, data-dependent weaknesses in software, application design flaws, and implementation-related vulnerabilities. The last chapter applies the techniques to Windows media player 9.0, Mozilla web browser 1.2.1, and OpenOffice.org 1.0.2 (Linux). The CD-ROM contains two testing tools written at the Florida Institute of Technology. (Annotation © 2004 Book News, Inc., Portland, OR)

Winkler, Ira. *Corporate Espionage*. Prima Publishing, 1997 (ISBN 0761508406).

A former agent for the National Security Agency reveals how American companies are losing billions of dollars each year through espionage perpetrated by computer hackers and employee spies and shows how companies can defend against it. (from the WorldCat Database)

Zuse, Horst. *Software Complexity: Measures and Methods* (Programming Complex Systems, No. 4). Berlin, Germany: Walter de Gruyter, Inc., 1991 (ISBN 311012226X).

Presents the knowledge and tools necessary for critically evaluating existing and future software complexity measures, understanding software measurement and theory, and applying standardized software complexity measures in practice. After a theoretical discussion, some 90 software complexity measures and their application to software complexity measurement are presented. (Annotation © Book News, Inc., Portland, OR; booknews.com)

Fields

Name	Value
Publication Date	12/09/06
resource-categories	books